محاضرة ١٠: Sharpening

المحتوى: ① مقدمة

② Derivative Operator
③ Laplacian Operator
④ Unsharp Masking and Highboost filtering
⑤ Sobel filtering
⑥ Combining spatial Enhancement Methods

مقدمة

* المحاضرة دي فيها رياضيات كتير (مش صعبة)، هحاول أشرحها بشوية تفصيل بس قدر الخص.
* هنعرف الأول الـ Derivative operator وبعدها هنستخدمه في الـ Laplacian
* بعدها هنعرف الـ Unsharp Masking وازاي نقدر نستخدمه في الـ Sharpening
* بعدها هنعرف طريقة الـ non-Linear للـ Sharpening
* وأخيرا، هنلم اللي عرفناه من طرق الـ Enhancement في تطبيق واحد، مش هنطبق بأرقام بس هنشرح آخر سلايد [10.13]

* فاكرين الـ Smoothing

Smoothing = blurring = Averaging/Integration = Low pass filtering

(spatial)

It was mostly done by linear operation (sum of products like Correlation and Convolution) or non-linear (median filter)

Sharpening = spatial differentiation = High pass filtering

Can be done by linear operations (Laplacian operator is considered applied by sum of products) or non-Linear

* أرجح جدا اللي ميفهمش الجزء ده يذاكره من المرجع من صفحة 179 لصفحة 194

---

* فكرة الـ Sharpening هي إن لما يبقى فيه 2 بكسل، والفرق الـ intensity بينهم كبير، فغالبا عشان دول حافة مختلفين، وبالتالي عايز نطلع الـ edges اللي بينهم ونوضحها أكتر

* ملاحظة مهمة : ال sharpening بيوضح ال noise كمان لو كانت موجودة و البكسلز اللي فرق ال intensity بينهم مش كبير (زي البكسلز المتجاورة) بيقلل وضوحها.

* هنستعمل ال first-order derivative وال Second-order derivative [هنطبقها بشكل Numeric عشان الصورة Digital]

Derivitive Operators

* هنعرف شوية شروط للمشتقة الأولى والتانية لما تبقى discrete ، هنشوف المعادلة بتاعتها ونشرح مثال لدالة 1-D ، وبعد كده نستخدم الكلام ده في 2D image بال Laplacian Operator [زي ال Laplacian filtering]

* راجع سلايد [10.3] ، راجع سلايد [10.4] بعد ماتذاكر الجزء ده

* شروط المشتقة الأولى :

① تدي صفر في المناطق اللي ليها Constant intensities

② متديش صفر في بداية ال step وال ramp

③ متديش صفر في ال ramp لما يكون ميلها ثابت

* شروط المشتقة التانية :

① تدي صفر في المناطق اللي ليها Constant intensity

② متديش صفر في بداية ونهاية ال step وال ramp

③ تدي صفر في ال ramp لما يكون ميله ثابت

[onset = بداية]

حساب المشتقة الأولى لدالة f(x) والمشتقة التانية (تفاضل جزئي)

$$1st\ order = \frac{\partial f}{\partial x} = \left.\frac{df}{dx}\right|_{x\,(1D)} = f(x+1) - f(x)$$

هنسميهم look-ahead operators

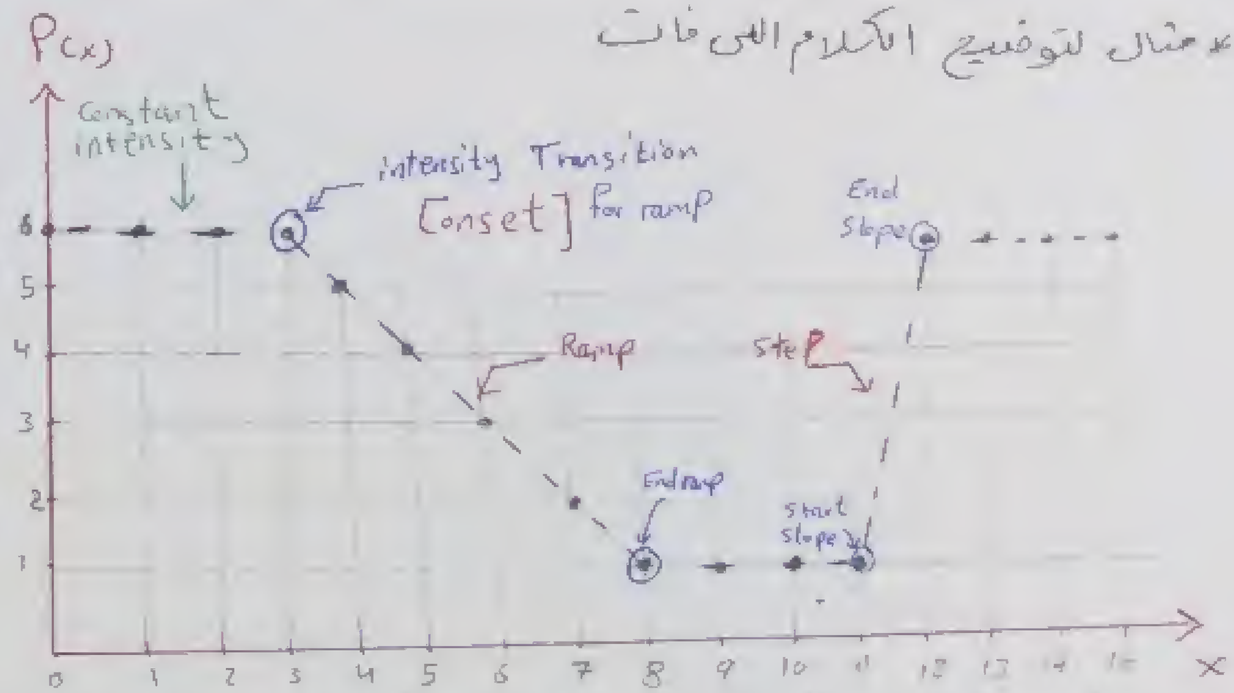
وافتكر انك بالنسبة ل 1st derivative عشان هنبص قدام

التفاضل الجزئي هو نفس التفاضل الكلي عشان الدالة في x بس

$$2nd\ order = \frac{\partial^2 f}{\partial x^2} = \left.\frac{d^2 f}{dx^2}\right|_{x\,(1D)} = f(x+1) + f(x-1) - 2f(x)$$

* مثال لتوضيح الكلام اللي فات

Discrete القيم والخط المنقط للتوضيح ال ramp وال step وال Constant



1st Derivative at $x = 6 = f(x+1) - f(x) = f(7) - f(6) = 2 - 3 = -1$

2nd Derivative at $x = 11 = f(x+1) + f(x-1) - 2f(x) = f(12) + f(10) - 2f(11) = 6 + 1 - 2 = 5$

اعمل الجدول بنفس الطريقة

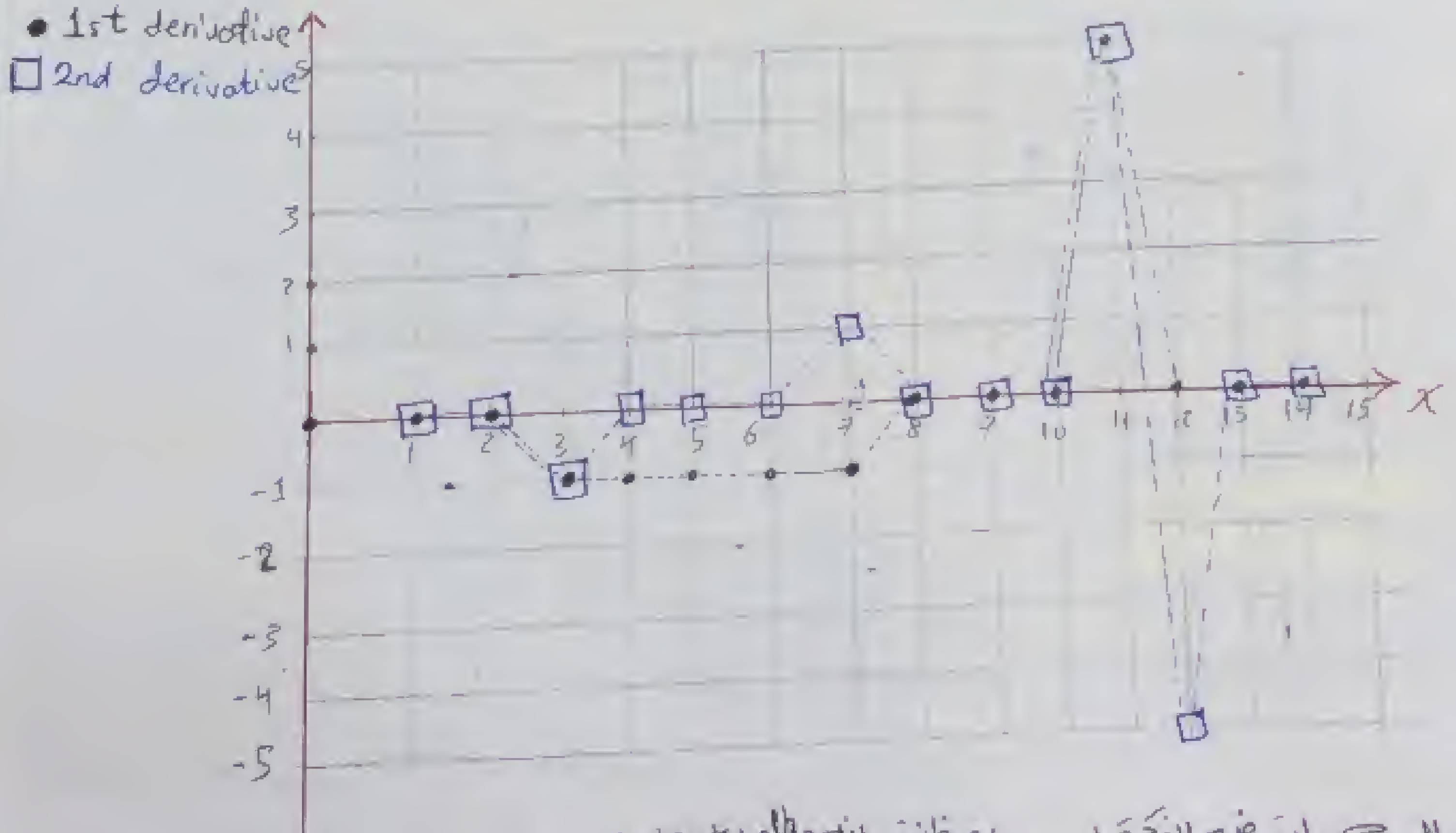
	x=0	x=1	x=2	x=3	x=4	x=5	x=6	x=7	x=8	x=9	x=10	x=11	x=12	x=13	x=14	x=15
1st Derivitve	0	0	0	-1	-1	-1	-1	-1	0	0	0	5	0	0	0	
2nd Derivative		0	0	-1	0	0	0	0	1	0	0	5	-5	0	0	

مفيش f(-1)

مفيش f(16)

* هنرسم ال Derivative مع ال x عشان نشوف شكلها عامل إزاي

* راجع الشروط اللي قولناها وشوفنها عالرسم (مهم أوي تفهمها)



الرسم لتوضيح الفكرة بس ، مظنش انه مطلوب بحي هامة

Laplacian operator

* هنوظف اللي عرفناه في ال 2D ، هنعمل ال 2nd Derivative مرة لمحور x ومرة لمحور y وهنكون الصورة عندي اسمها f(x,y)

for x: $\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x,y)$

for y: $\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x,y)$

هنجمع العمليتين في عملية واحدة اسمها ⟵ $\nabla^2 f(x,y)\Big|_{(x,y)}$

$$\nabla^2 f(x,y)\Big|_{(x,y)} = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

(x,y) ⟵ المجاور اللي نتعامل عليها

شكل ال Mask (هاخد ال Coefficients من معادلة $\nabla^2 f(x,y)$)

x-mask + y mask = Laplacian Mask (A)

x-mask ($f(x-1,y)$ above, $f(x,y)$ center, $f(x+1,y)$ below):

0	1	0
0	-2	0
0	1	0

+

0	0	0
1	-2	1
0	0	0

=

0	1	0
1	-4	1
0	1	0

* ممكن آخد ال pixels في الزوايا معايا ، وممكن أعكس الإشارة زي D و C

B (خدت البكسلز في الزوايا)

1	1	1
1	-8	1
1	1	1

C

0	-1	0
-1	+4	-1
0	-1	0

في حالة C, D هتتغير الإشارات في المعادلات

D

-1	-1	-1
-1	+8	-1
-1	-1	-1

علاحظنا انه عملية ال derivation عملية Linear وبالتالي ال Laplacian عملية Linear

* لاحظ إنه فلتر y هو فلتر x بس اتعمله rotation ب 90°

تبقى معادلة $\nabla^2 f(x,y)$ لو خدنا الفلتر (c)

$$\nabla^2 f(x,y) = f(x+1,y) - f(x-1,y) - f(x,y+1) - f(x,y-1) + 4\,f(x,y)$$

* لما هطبق ال Laplacian هيطلعلي صورة جديدة ، بأجمعها أو أطرحها من الصورة الأصلية

* هطرح لو المعامل ال $f(x,y)$ سالب (ال Center بتاع ال Mask) يعني ال Mask

* هجمع لو معامل ال $f(x,y)$ كان موجب (ال Center بتاع ال Mask) يعني ال Mask

* يبقى لما أجي أعمل Sharpening باستخدام ال Laplacian operator

$$g(x,y) = f(x,y) + c\left[\nabla^2 f(x,y)\right]$$

$g(x,y)$ ← الصورة بعد ال Sharpening

$f(x,y)$ ← الصورة الأصلية

c ← موجبة لو ال Center بتاع ال Mask موجب وسالبة لو العكس

$\nabla^2 f(x,y)$ ← الصورة اللي طلعت بعد عمل ال Laplacian Filter

* راجع سلايد [10 , 6]

* عكس ال Laplacian operator طلع قيم سالبة لل pixel intensity ، في الحالة دي هيعمل rescale and normalize اللي شرحناها في محاضرة 4 سلايد [4 . 10]

* راجع سلايد [10 . 7] ، فيها صور توضح الفرق لما تعمل من غير rescale and normalize

(سماها scaling ، اختلاف مسميات كتير ، بس ماتاخدش في بالك) ولما تعمل (Scaling) .

* ركز كويس جدا في المثال.

Unsharp Masking and Highboost filtering

* دي طريقة تانية عشان نعمل Sharpening

* افتح سلايد [10 . 9] ~~[illegible]~~ ، الفكرة سهلة جدا

* راجع سلايد [10.9] كويس لما تخلص الجزئية دي

الفكرة إنه هيستخدم ال blurring عشان يطلع ال edges، إزاي ؟؟!

- افرض صورة $f(x,y)$ ، وبعد ال blurring بقت $f'(x,y)$ (مجرد مسميات)
- هطلع ال Mask اسمه $g_{mask}(x,y)$ بطرح الصورة ال blurred من الصورة الأصلية

$$g_{mask} = f(x,y) - f'(x,y)$$

- هضرب ال Mask ده في factor اسمه k (ده هيسمحلي أتحكم في ال Sharpening شديد ولا لأ ، يعني هل يظهر ال edges أوي ولا لأ)

وأجمع ال Mask على الصورة الأصلية (خلي بالك ال Mask بأبعاد الصورة)

$$g(x,y) = f(x,y) + k\, g_{mask}(x,y)$$

* عندك مثال على الكلام ده على signal في السلايد [10.9]

* عشان يبقى فيه قيم pixels بالسالب بعد العملية فبنعمل scaling [rescaling & normalizing]

ملحوظة مهمة: ال filters أوقات بيقولوا عليها isotropic أو لأ

بمعنى إنه الفلتر Isotropic ، إنه هيدي نفس النتيجة لما تعمل rotation لل Mask بتاعك ، زي:

Isotropic

0	1	0
1	-4	1
0	1	0

Non-Isotropic

-1	2	-1
0	0	0
1	-2	1

Sobel filtering

* فلما فهمنا إزاي نحسب ال 2nd derivative سواء في ال y أو ال x وكتبناها على الشكل $\nabla^2 f(x,y)\big|_{(x,y)}$

* راجع سلايد [10،10] بعد الكلام ده. أو من المرجع بداية من صفحة 187 أحسن لو مفهمتش الكلام المكتوب.

* ال 1st derivatives عكس نطبقها من خلال ال Magnitude لل gradient وال gradient هو مجرد vector اللي كنا عملناه قبل كده بس مع اختلاف مسميات، عارف الكلام ده بالضبط، اعتبر اللي فات تعريف لل 1st derivative وده تعريف تاني في ضوء ال non-Linear Sharpening

باستخدام ال gradient

هنعرف ال gradient بأنه

$$\nabla f = grad(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

vector ال جو بقواقيم انهم يد الجد كل فهم، لكه كده قبل مرة حساهم ←

* للـ اتجاه ، وهنجيب ال Magnitude لل vector ده، وهنسمي ال Magnitude بـ M :

$$M(x,y) = mag(\nabla f) = \sqrt{g_x^2 + g_y^2}$$

$$= \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

* ال M هيبقى صورة بنفس أبعاد الصورة الأصلية لما أمشي على كل البكسلز

* ممكن نعمل Approximation لحساب M ويكون efficient في ال Computation

$$M(x,y) \simeq |g_x| + |g_y| \quad (*)$$

* فيه شوية ملاحظات:

1. ال vector اللي هو ∇f بيشاور على أكبر قيمة للتغير في قيمة f عند بكسل (x, y) وطبعا الـ Magnitude هو قيمة التغير ده (change rate value)
2. عملية ال gradient عملية Linear بس not isotropic
3. ال Magnitude = = non-Linear بس Isotropic
4. ال Magnitude في حالة المعادلة (*) بيكون non-Linear وكذلك non-Isotropic

* ليه الملاحظات اللي فاتت ؟

* عملية ال derivation كانت عملية linear عشان كده ال grad بردو Linear

* عملية ال Magnitude عباره عن تربيع للي تحت الجذر , أو قيمة مطلقة عشان كده مش linear

* عشان عملية ال grad عملية Linear فممكن نعملها في شكل Sum of product من خلال ال Masks التالية

المنطقة 3x3 اللي هشتغل عليها في الصورة

$$\begin{matrix} z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 \\ z_7 & z_8 & z_9 \end{matrix}$$

Roberts Operators

x:

-1	0
0	1

y:

0	-1
1	0

* عشان هشتغل على ال center بتاع الصورة اللي هو (z_5)

* لاحظ الفلتر غريب شوية بس كان كده أول ما طلع في 1965 باسم اللي عمله (Robert)

الناتج هيبقى بالشكل التالي (بعد ال Mask على المنطقة المحددة بالأخضر)

$$g_x = (z_9 - z_5) \quad \& \quad g_y = (z_8 - z_6)$$

وهتبقى ال gradient image اللي هي $M(x,y)$ عند z_5 (يعني أصبح ال pixel)

$$M(x,y) = \left[(z_9 - z_5)^2 + (z_8 - z_6)^2 \right]^{1/2}$$

OR more effeciently for computation

$$M(x,y) \approx |z_9 - z_5| + |z_8 - z_6|$$

ومع بعد كده نعدل على شغل Roberts و نحول ال Mask بتاعك لحاجة 3x3 ، أسهل في الشغل من ال 2x2 بتاع Roberts و دي اللي بنتعامل بيها من الأول

قبل ما نعدل عليه نشوف إزاي ال 1st derivative بمفهومها الأول (هنسميه look-ahead operators)

هنجيب g_x , g_y

$$\begin{matrix} z_1 & z_2 & z_3 \\ z_4 & z_5 \xrightarrow{y+1} & z_6 \\ z_7 & z_8 & z_9 \end{matrix}$$

$$\boxed{\begin{aligned} g_x &= z_8 - z_5 \\ g_y &= z_6 - z_5 \end{aligned}}$$

ولو حسبنا ال M بنفس الطريقة فوق

احنا كنا عارفين المعادلة

(1-D) $$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

(2-D) $$\frac{\partial f}{\partial x} = f(x+1,y) - f(x,y)$$

Roberts operators مع Mask 3x3 عمل Sobel operator جه بقى ال

original

$$\begin{array}{ccc} z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 \\ z_7 & z_8 & z_9 \end{array} \qquad \begin{array}{c} x \\ \begin{array}{ccc} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{array} \end{array} \qquad \begin{array}{c} y \\ \begin{array}{ccc} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{array} \end{array}$$

اللي تحت ناقص اللي فوق

اليمين ناقص الشمال

$$g_x = \underbrace{(z_7 + 2z_8 + z_9)}_{\text{اللي تحت}} - \underbrace{(z_1 + 2z_2 + z_3)}_{\text{اللي فوق}}$$

$$g_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

$$M(x,y) \simeq |g_x| - |g_y|$$

$$\simeq |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)|$$

$$+ |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|$$

راجع بقى ال Slide رقم [10.11] بعد الكلام ده ومعاها [10.10]

* تطبيق على ال Sobel filtering في [10.12]

الصورة على شمال عدسات عيون ، وفيها مشكلة ، عمل عليها Sobel filter

طلع فيه صورة x و صورة y و عمل Magnitude ليهم جاب الصورة على اليمين ، مشان منها الظل في الزوايا وخط انطباع الدايرة

** Sobel filtering is good for automated inspection due to simplified computation task **

Combining Spatial enhancement methods

في سلايد [10.13] شارح تطبيق عشان يحسن صورة ، هيعديها على كذا عملية لحد مايوصل للصورة h من الصورة a

* وبيوريك في نفس الفرق بين لو استخدمت Laplacian أو Sobel مثل

a	b
c	d

e	f
g	h

* هيعمل العمليات للهابدون scaling وهيعمله في الآخر طالما يخلص كل العمليات
* الخطوات اللي عملها (أرجع جدا اقرأ صفحة ١٩١ و ١٩٤ في المرجع)

① في البداية هو عايز يطلع details أكتر من الصورة

② عمل Laplacian operator طلعت الصورة (b) ، جمعها على (a) وطلع (c)

③ منه ملاحظ في (c) إنها noisy ، فعشان يتخلص من الـ noise هيعمل Smoothing بس بلاش الـ median ، هيضيع الـ features

④ فالك تعالى نعمل Sobel ونجيب (d) هيبقاش noise كتير وهكذا هستعملها كـ Mask للصورة (c) ، الـ Sobel بيجيب منه الـ gradient

⑤ هيعمل Smoothing للصورة بتاعت الـ gradient ويطلع (e) وبعد كده يضربها كـ Mask في صورة الـ Laplacian image فيعمل $c * e$ ويطلع (f)

⑥ كده بقت الـ edges عندي ظاهرة جدا ، أضيفها على الصورة الأصلية (a) وأطلع الصورة (g)

⑦ هيعلي الـ Dynamic Range من خلال الـ power Law أو الـ Gamma Correction

(ممكن ترجعها في سلايد (6.12) ، مفروض $\gamma=0.5$ و $c=1$ وجاب الصورة (h)

* كان ممكن نعمل الخطوة ⑦ بـ Histogram specification ، بس الـ power Law أسرع في التطبيق ده تحديدا

* هكرر تاني ، أنصح بشدة المثال ده يتراجع من المرجع صفحة 191, 194